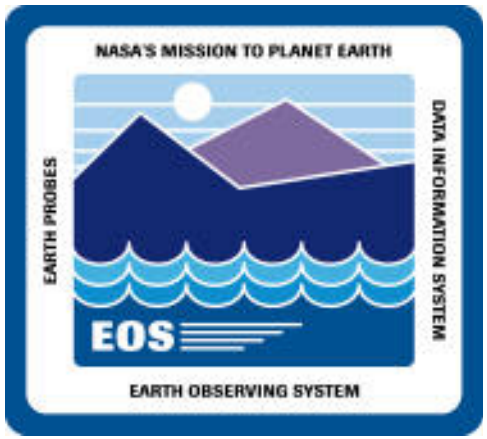
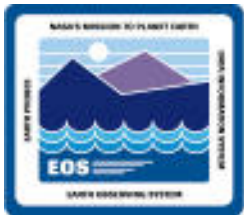


DAAC Unique Extensions

Carl Solomon

31 October 1995

Overview



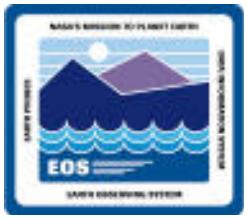
DAAC Unique Extensions provide capabilities for DAACs to graft their own services to ECS provided components.

Provides interoperability of DAAC systems with ECS.

Application Programming Interfaces (APIs) provide mechanism to accomplish extensions.

- **APIs consist of public operations on public classes in C++**
  - **a public class is exported by a CSCI for use in another**
- **Same classes as being used by ECS software**

Application Programming Interfaces (APIs)



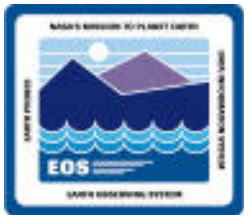
Public APIs documented in DID 313-CD-006-001 - Release B CSMS/SDPS Internal Interface Control Document for ECS Project

- **At IDR, the public operations and attributes are identified and described in DID 313.**
- **Additionally, at CDR, the public operations will have signatures (calling sequences) documented in DID 313.**

Description of design encapsulated by APIs documented in the respective DID 305 Design Documents

For CDR, a preliminary version of an API Interface Description Document (IDD) will be developed to help describe use of public classes.

Development Environment Required



The following development tools at a minimum would be required at the DAAC to extend the ECS functionality:

- **C++ compilation and link Tools on server platforms**
  - **to graft DAAC unique functionality to ECS source code**
- **OODCE/DCE Development Environment**
  - **to support Distributed Object Framework needed by ECS s/w**
- **COTS C++ Libraries (RogueWave)**
  - **objects required, source optional to help in understanding use of common library s/w**
- **ECS Class Libraries**
  - **to link in with DAAC unique s/w**

On ECS platforms, these tools available in ECS M&O environment

- **need to account for simultaneous user and licensing issues**

Layered Environment

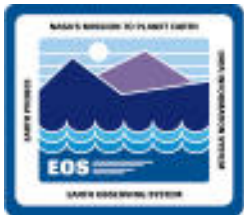
Development of DAAC Unique Extensions requires understanding of:

- **ECS Architecture,**
- **applicable ECS subsystem(s),**
- **C++ source, and**
- **C++ compilation and link environments**

<table>
<tr><td colspan="3">DAAC Unique Extensions</td></tr>
<tr><td colspan="3">ECS Software</td></tr>
<tr><td rowspan="2">X-windows</td><td>OODCE</td><td rowspan="2">RogueWave</td></tr>
<tr><td>DCE</td></tr>
<tr><td colspan="3">Operating System</td></tr>
</table>

ECS development environment maps to layered architecture.

Example DAAC Unique Extension



Desire: DAAC wishes to integrate DAAC unique analysis s/w, that creates a new type of product, directly with ECS Science Data Server. Analysis s/w is desired to insert and acquire datasets into the Science Data Server.

Step 0: Obtain Appropriate Approvals from ESDIS and analyze performance and sizing impacts. Insure hardware supports desired extension.

Step 1: DAAC acquires required development environment.

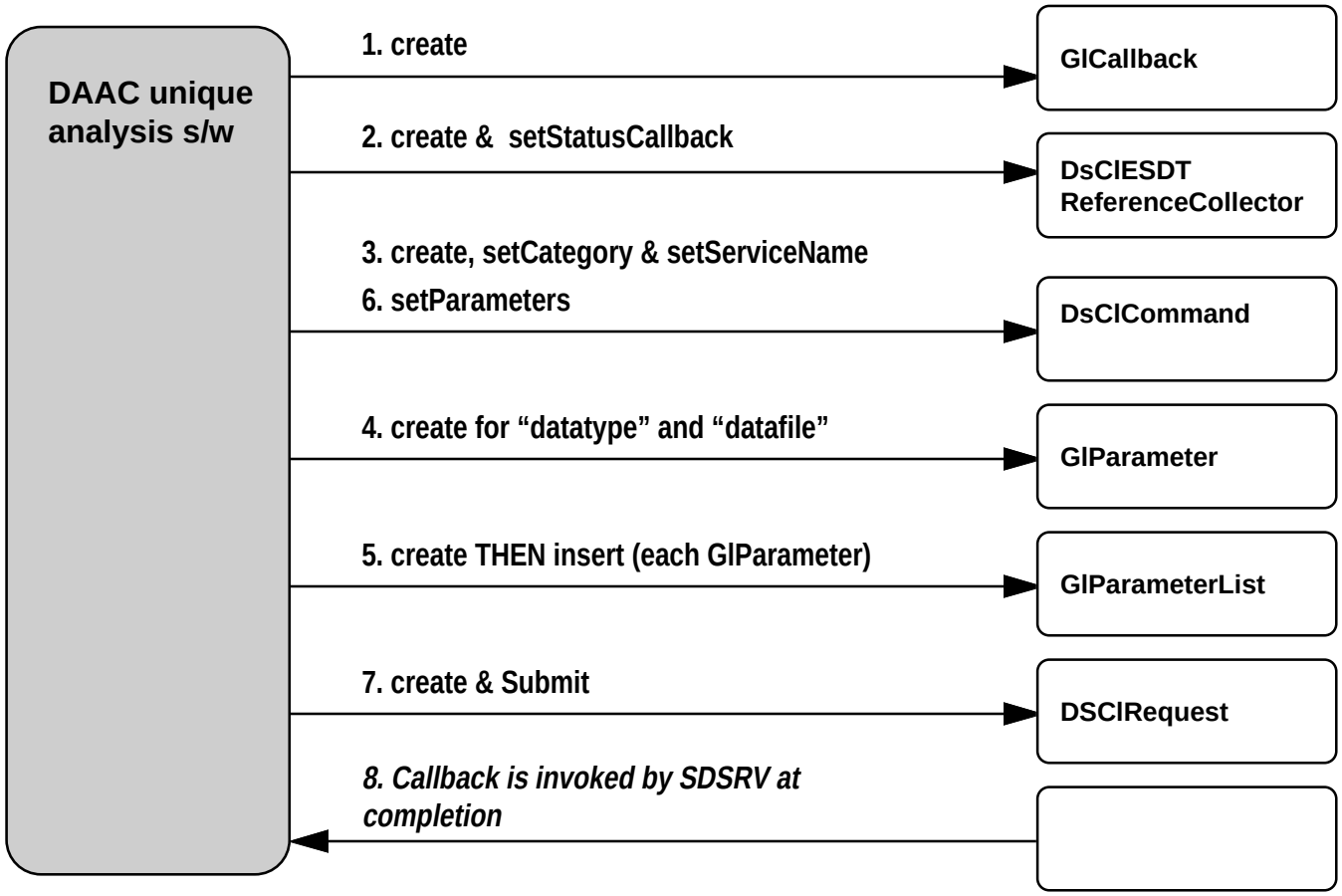
Step 2: DAAC s/w developers use documentation to identify public interfaces (APIs) required to accomplish integration.

Step 3: Make any new ESDT known to Science Data Server by creating a Descriptor and data type implementation for inclusion in Science Data Server. *(Storage of new ESDT in Science Data Server would need to be Sized.)*

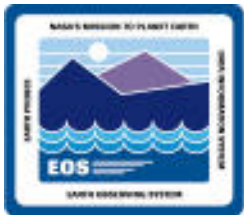
Modification to Insert Datasets Into Science Data Server

Step 4: Modify DAAC unique analysis s/w to make calls to the public interfaces provided by Science Data Server to perform dataset insertions.

- ***Need to interface to SDSRV and common global classes to complete task.***



Modification to Acquire Datasets from Science Data Server



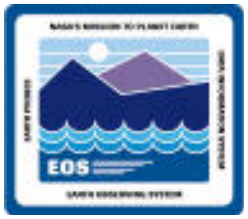
Step 5: Modify DAAC unique analysis s/w to make calls to the public interfaces provided by Science Data Server to perform dataset acquires.

Step 6: Place ad using Advertising service on new product.

DAAC unique analysis s/w is able to generate new product in the Science Data Server

New product is available to users.

Next Steps/Plan for CDR



Further refinement of APIs as signatures are defined in DID 313.

Preliminary API IDD describing use of public classes to accomplish approximately 10 representative DAAC unique extensions (e.g. DAAC unique Ingest Services, DAAC unique Data Distribution Services).

- **Preliminary: Release B CDR (April 1996)**
- **Final: Release B CDR + 4 months (August 1996)**